

メモリアクセスの時間的周期性を利用した動的データプリフェッチ機構

3 F-4

安島雄一郎, 中村友洋, 田中英彦

東京大学工学部

1 はじめに

近年、マイクロプロセッサの高速化に伴い、相対的にメモリアクセスのレイテンシが大きくなってきた。このため、メモリアクセスのレイテンシがマイクロプロセッサの高速化に大きな障害となってきた。従来、キャッシュシステムによってレイテンシが削減されてきたが、キャッシュミスは全メモリ参照の1~2割程度存在するため十分とはいえない。本研究ではさらなるメモリアクセス・レイテンシの隠蔽を目指し、メモリアクセスの時間的周期性に着目した新しい動的データ・プリフェッチ機構を提案し、その性能を評価する。

2 動的データ・プリフェッチ

データ・プリフェッチを行なうには、参照するアドレス、プリフェッチを行なう時期を決定する機構が必要である。動的データ・プリフェッチではメモリ参照の履歴などの実行時情報からアドレス予測、プリフェッチを行なう。

2.1 参照アドレスの予測

参照アドレスが変化する間隔(ストライド)は一定となる確率が高い[3]。このため、ストライド情報を利用する参照アドレス予測方法(以下ストライド法)が提案されている[1][2][3]。

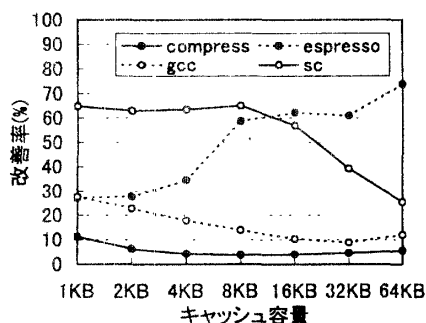


図1: キャッシュミスの改善率

図1は、整数系の一般プログラムで代表的な4つのアプリケーションにおいて、キャッシュミスのうちストライド法でアドレスが予測できるものの割合を表したグラフである[3]。これより、ストライド法を用いた動的データプリフェッチ機構によって、メモリアクセスのレイテンシをさらに隠蔽することが可能であることが分かる。

Dynamic Data-Prefetching method using temporal interval information

Yuichiro AJIMA, Tomohiro NAKAMURA, Hidehiko TANAKA

Faculty of Engineering, University of Tokyo

2.2 プリフェッチ・タイミングの決定

Bear, Chen による Lookahead data prefetching では、プリフェッチ・タイミングの決定に Lookahead-PC(LA-PC)を使用する[1]。LA-PCはプログラムカウンタ(PC)に先行して命令フェッチを行なっており、LA-PCにロード命令がフェッチされるとプリフェッチが行なわれる。LA-PCは常にPCからメモリアクセスのレイテンシに相当する命令数だけ先行しており、これによって最適なプリフェッチ・タイミングが生成される。しかし、この方式には以下のようにいくつかの大きな欠点がある。

- ・フェッチ機構を二重化するためハードウェア量大
- ・PCによる本来の命令フェッチの妨げとなる
- ・分岐予測機構に性能を大きく依存する

このため、Lookahead data prefetchingの実装によって大幅な性能向上を期待することは難しい。

3 新しい動的データ・プリフェッチ機構の提案

ここで、実装が比較的容易なプリフェッチ・タイミング生成機構を用いた動的データ・プリフェッチ機構を提案する[3]。提案する機構では、ロード命令実行時に将来の参照アドレスとロード命令実行時期を予測し、予測情報をバッファに保存。目標の時期にプリフェッチを行なう。

3.1 インターバル法

ロード命令実行時期の予測法として、インターバル法を提案する。これは、参照アドレスの予測にストライド法を使用していることを前提とする方法である。

図2は各ロード命令がストライドアクセスを行なう場合、そのロード命令が再び実行されるまでの命令数間隔を調べ、その変化の分布を表したものである。これより、実行命令数間隔の変化は0である場合が多いことが分かる。よって、ストライド法でアドレス予測が成功する場合、ロード命令の実行命令数間隔も一定である確率が高いといえる。インターバル法ではこの性質を利用し、ストライド法と同様、ロード命令の実行間隔(clock単位)が一定であると仮定して予測を行なう。

3.2 繰り返し倍率

ロード命令の実行間隔がメモリアクセス・レイテンシより小さい場合、常に次のロード命令だけに対する予測を行っていたのではプリフェッチが間に合わない。このため、ロード命令の実行間隔が小さい場合は複数回先の実行を予測し、プリフェッチする必要がある。

ストライド法、インターバル法で n 回先の予測を行なうには単純に間隔情報を n 倍すれば良い。以下ではこの n を繰り返し倍率と呼ぶ。提案する機構では予測情報と

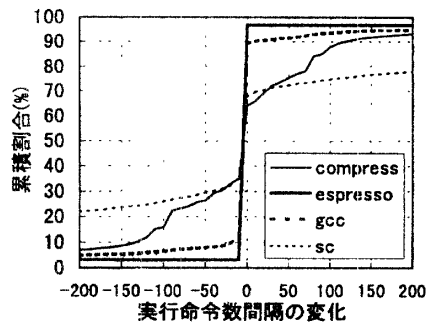


図 2: 実行命令数距離誤差の分布

して繰り返し倍率を保存する。初めは倍率1に設定するが、一回分の実行間隔ではレイテンシを十分に隠蔽できない場合、プリフェッチを即座に実行した上で倍率を2に更新、さらに倍率2の予測情報をプリフェッチバッファに送る。以降同様に、レイテンシを隠蔽できる大きくなるよう、繰り返し倍率を動的に調整する。

4 評価

前章では実行命令数距離を根拠に提案を行なったが、提案した機構はプロセッサが実際に動作するクロックを用いて予測を行なう。実際のプロセッサでは様々な要因でIPC(Instruction Per Cycle: 1サイクルの命令実行数)が変動するため、実際のプロセッサに実装した場合のメモリシステムの性能向上を評価するには、シミュレーションを行なう必要がある。

本研究ではスーパースカラ・プロセッサのシミュレータを実装し、プリフェッチを導入した場合としない場合の実行速度を比較することで評価を行なった。メモリアクセスのレイテンシを1とした理想的な場合の実行サイクル数(C_i)を目標値、プリフェッチを導入しなかった場合の実行サイクル数(C_n)を基準値とし、プリフェッチを導入した場合の実行サイクル数(C_p)よりメモリシステム性能の改善率(%)を次のように算出する。

$$(C_n - C_p) \div (C_n - C_i) \times 100$$

プログラムは代表的な整数系一般プログラム4種を使用し、各条件で1000万命令のシミュレーションを行なった。また、繰り返し倍率を導入した場合と導入しない場合の評価も行なった。

5 結果及び考察

シミュレータによる評価の結果を図3,4に示す。図3は繰り返し倍率を1に固定した場合、図4は繰り返し倍率を可変にした場合である。

全体的な傾向として、キャッシュサイズが小さい場合は性能が悪くなっている。これは、プリフェッチによって必要なデータが追い出されてしまうためと考えられる。また、プログラムによって性能が改善されたものもあるが、逆に悪くなったものもあるが、繰り返し倍率を用いた場合その差が大きくなっている。ここで、繰り返し倍率を導入した場合はストライドアクセスが途切れた時に余計なプリフェッチがバースト的に発生することを考慮す

ると、性能低下の原因は不要なプリフェッチの影響が大きいと考えられる。

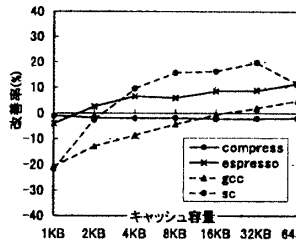


図 3: 倍率1固定

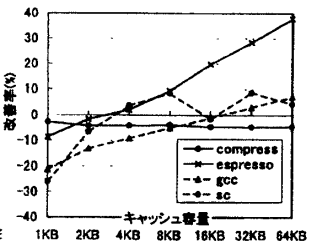


図 4: 倍率可変

そこで、不要なプリフェッチによる影響を緩和するため、プリフェッチによるメモリアクセスは、ロード命令やストア命令によるメモリアクセスによって中止されるものとした。この時の結果を図5,6に示す。

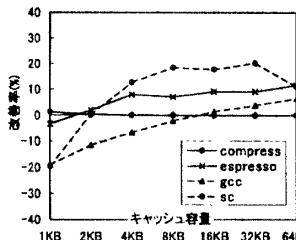


図 5: 倍率1固定(改)

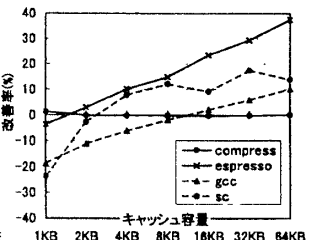


図 6: 倍率可変(改)

メモリアクセスの中止を許した場合、全般的にメモリシステムの性能が向上しているが、特に繰り返し倍率を可変にした場合に効果が大きいことが分かる。

6 まとめ

提案した動的データプリフェッチ機構により、キャッシュで削減できなかったレイテンシをさらに0~4割程度削減できることが分かった。また提案した機構では、読み出しの中止を許した場合に最大の性能が得られることが分かった。

今後の研究課題として、本研究で提案した機構を実装するのに必要なハードウェア量についての定量的な評価が挙げられる。また、ストライド法は論理アドレス空間上の予測であり、仮想メモリ空間ごとに予測情報を独立させるなどの応用が考えられる。

本研究の一部は文部省科学研究費(一般研究(B) 課題番号 07458052「大規模データバスプロセッサの研究」)による。

参考文献

- [1] Tien-Fu Chen, Jean-Loup Bear: "A Performance Study of Software and Hardware Data Prefetching", Proc. 21st Annual Symposium on Computer Architecture, pp.223-232 (1994).
- [2] 勝野 昭, 木村 康則: "ロード・アドレス予測方法の検討", 情報処理学会研究会報告 ARC117-9 (1996).
- [3] 安島 雄一郎: "動的データプリフェッチを用いたメモリシステムに関する研究", 東京大学工学部 卒業論文, <http://www.mtl.t.u-tokyo.ac.jp/Research/paper/1996/J96-thesis-aji-1.ps.gz> (1997).